

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS  
CURSOS

FORMACIONES

CURSOS

PARA  
EMPRESAS

ARTÍCULOS DE TECNOLOGÍA &gt; DATA SCIENCE

# ¿Append o Extend? Agregar elementos a la lista con Python



Alex Felipe

13/10/2020

En el desarrollo de cualquier sistema es muy común tratar con **listas**, como una lista de productos, por ejemplo. Supongamos que tenemos la siguiente representación de un producto:

```
class Producto (objeto): def __init__ (self, nombre, valor):
    self.__ nombre = nombre
    auto __ valor = valor

def __repr__ (self):
    return "nombre:% s valor:% s"% (self.__ nombre, self.__ valor)

def get_nombre (self):
    return self.__ nombre

def get_valor (self):
    return self.__ valor
```

Entonces tenemos una lista con algunos productos:

```
from model import Producto

Chocolate = Producto("Chocolate", 4.35)
Jugo = Producto("Jugo", 3.75)
Frijoles = Producto("Frijoles", 10.5)
Producto = [Chocolate, Jugo, Frijoles]

print Productos
```

Si ejecutamos este código, tenemos el siguiente resultado:

```
[nombre:Chocolate valor:4.35, nombre:Jugo valor:3.75, nombre:Frijoles valor:10
```



Por ahora, todo va como se esperaba. ¿Pero qué pasa si queremos agregar un nuevo producto a esta lista? ¿Cómo podemos hacer eso? En las listas de **Python**, hay dos funciones capaces de **agregar elementos**, que son **append** y **extend**. ¡Entonces probemoslos! Comenzaremos con el **append**:

```
producto4 = Producto("Queso", 4.25)
productos.append(producto4)

print productos
```

Mira cuál es el resultado:

```
[nombre:Chocolate valor:4.35, nombre:Jugo valor:3.75, nombre:Feijao valor:10.5
```



Aparentemente todo salió como se esperaba. Entonces, probemos la función **extend**:

```
producto4 = Producto("Queso", 4.25)
productos.extend(producto4)
```

```
print productos
```

Probamos el código nuevamente:

```
TypeError: 'Producto' object is not iterable
```

¡Uy! Parece que extender no puede agregar un objeto a la lista ... ¿En serio? Si analizamos el mensaje de error, nos informa que el objeto de tipo "Producto" no es **iterable**, pero ¿qué significa eso?

extend, a diferencia de append, itera sobre el elemento que desea agregar, es decir, involucra el argumento pasado por parámetro dentro de un ciclo, y luego agrega los valores contenidos dentro de ese parámetro, en este caso, intentó hacer esto con el objeto del tipo Producto. Esta es precisamente la razón por la cual no fue posible agregar el objeto de tipo Producto y presentó este error.

*Entonces, ¿eso significa que no podemos usar extend para agregar objetos?*

¡No exactamente!

Una solución a este problema es, en lugar de enviar un objeto puro, enviar como parámetro una **estructura iterable** que contenga un objeto de tipo "Producto", es decir, envíe una lista que contenga un Producto!

```
producto4 = Producto("Queso", 4.25) productos.extend([producto4])
```

```
print productos
```

¿Funciona ahora? Veamos el resultado:

```
[nombre:Chocolate valor:4.35, nombre:Jugo valor:3.75, nombre:Frijoles valor:10
```



Funcionó como se esperaba! Sin embargo, ¡solo hemos agregado un solo elemento! ¿Qué pasa si quisiéramos agregar varios elementos? En otras palabras, agregar una lista de

nuevos productos. ¿Funcionaría sin problemas? Probemos primero con el append:

```
producto_nuevo1 = Producto("Jugo", 4.5)

producto_nuevo2 = Producto("Pan", 3.0)

productos_nuevos = [producto_nuevo1, producto_nuevo2]

productos.append(productos_nuevos)

print productos
```

Al probar el código obtenemos el siguiente resultado:

```
[nombre:Chocolate valor:4.35, nombre:Jugo valor:3.75, nombre:Frijoles valor:10
```

**Aparentemente funcionó**, pero observe que en lugar de agregar cada producto a la lista de productos, ¡agregó una lista completa! Es decir, cuando usamos append simplemente agrega el **valor completo a la lista**. Pero, ¿y si usamos extend? ¿Funciona? Es decir, ¿agrega producto por producto? Vamos a revisar:

```
producto_nuevo1 = Producto("Jugo", 4.5) producto_nuevo2 = Producto("Pan", 3.0)

productos_nuevos = [producto_nuevo1, Producto_nuevo2]

productos.extend(productos_nuevos)

print productos
```

Ejecutando el código nuevamente

```
[nombre:Chocolate valor:4.35, nombre:Jugo valor:3.75, nombre:Frijoles valor:10
```

Funcionó como se esperaba! Por lo tanto, concluimos que las dos funciones funcionan de la siguiente manera:

- **Append:** Agrega cualquier valor **completo**, por ejemplo, si enviamos un objeto, agrega el objeto, si enviamos una lista, agrega la lista completa en lugar de sus elementos.
- **Extend:** agrega elementos de una **estructura iterable**, por ejemplo, si enviamos un objeto puro, no sabe cómo agregarlo, sin embargo, si enviamos ese mismo objeto dentro de una lista, escaneará la lista y agregará ese objeto y, si hay otros, el resto dentro de la lista.

Podemos ver claramente la diferencia entre las dos funciones cuando intentamos agregar una string, por ejemplo, la string "chocolate":

- **Append:**

```
palabras = []  
  
palabras.append("chocolate")  
  
print palabras
```

Resultado del append:

```
['chocolate']
```

- **Extend:**

```
palabras = []  
  
palabras.extend("chocolate")  
  
print palabras
```

Resultado del append:

```
['c', 'h', 'o', 'c', 'o', 'l', 'a', 't', 'e']
```

Tenga en cuenta que, mientras `append` agrega toda la cadena, `extend` agregó letra por letra, porque la string, detrás de escena, es una array de **char**, por lo que escanea esta array, es decir, esta estructura iterable, y agrega cada artículo.

¿Qué tal aprender más sobre **Python** y sus diversos recursos? Entonces, ¡Mira nuestros cursos de **Python para Data Science** aquí en [Alura](#)!

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

**En Alura encontrarás variados cursos sobre Data Science.  
¡Comienza ahora!**

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

**NAVEGACIÓN**

PLANES



INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |  
Liderazgo y Gestión de Equipos | Startups y Emprendimiento